IBM Project Report

# Title: Data Backup and Recovery using Python

## Introduction

Data is one of the most critical assets in any organization. The loss of essential data due to system failure, human error, or malicious attacks can result in significant operational and financial consequences. This project presents a mini backup and recovery system built using Python, simulating key principles used in enterprise tools like IBM Tivoli Storage Manager.

## Objective

To design and implement a simple, automated tool that:  
- Backs up files from a specified directory.  
- Compresses the backup into a ZIP archive.  
- Allows recovery (restoration) from the backup archive.

## Tools & Technologies

- Programming Language: Python 3.x  
- File Compression: zipfile module  
- Configuration: JSON file  
- Platform: OS-independent (Windows/Linux/macOS)  
- Version Control: GitHub

## System Requirements

- Python 3.x installed  
- Basic understanding of directory structure  
- File system access permissions

## Methodology

1. Configuration Setup:  
 A config.json file defines:  
 - Source folder to back up  
 - Destination folder for backups  
 - Filename of the backup archive  
  
2. Backup Script (backup.py):  
 - Reads paths from config.  
 - Walks through the source directory.  
 - Adds all files to a ZIP archive.  
 - Logs the backup operation with a timestamp.  
  
3. Restore Script (restore.py):  
 - Reads the backup archive.  
 - Extracts its contents to the source directory.  
 - Ensures original file structure is preserved.  
  
4. Logging:  
 - A simple backup.log records each backup event.

## Project Structure

ibm-data-backup/  
├── backup.py  
├── restore.py  
├── config.json  
├── README.md  
└── backup.log (generated during use)

## Sample Configuration (config.json)

{  
 "source": "data/",  
 "backup\_dir": "backups/",  
 "backup\_name": "backup.zip"  
}

## Results

- The project was tested on test folders containing multiple files.  
- Backup and restore operations completed successfully.  
- ZIP archives created were consistent in size and structure.

## Conclusion

This project demonstrates the core functionality of a data backup and recovery system using Python. It mimics the automation and reliability of larger enterprise systems in a minimal and easy-to-understand form. This prototype can be extended further by integrating cloud storage or scheduled automation.

## Future Scope

- Add incremental backup support  
- Integration with cloud storage (e.g., IBM Cloud Object Storage)  
- Encryption of backup archives  
- Graphical User Interface (GUI)

## GitHub Repository

📁 (https://github.com/Swabira/Ibm-data-backup)